Amendments to the Specification

Please amend the specification as follows.

Please replace paragraph 9 with the following:

[0001] The bit error rate tester of the present invention may be implemented by programming programmable logic circuitry to implement transmission circuitry that communicates test data across the interface being tested and comparison circuitry that receives the test data from the interface's output. The output data is compared to comparison data, which corresponds to the original test data in order to determine whether any errors occurred to the data as it was communicated across the interface. Control circuitry, which may be in the form of a software-programmed processor, may also be used to control the operation of the transmission and comparison circuitry. User equipment, such as a personal computer, may be coupled to the bit error rate tester in order to provide a user with testing information and in order to provide the user with any suitable level of control over the testing procedure.

Please replace paragraph 15 with the following:

[0002] The present invention provides a more efficient bit error rate tester implemented using a programmable logic device. In particular, one or more programmable logic devices may be programmed to implement the features of a bit error rate tester in order to verify the bit error rate of that programmable logic device's one or more interfaces or of any other suitable interface. For example, a bit error rate tester implemented on or more programmable logic devices may be used to test the bit error rate of an interface that is distinct and independent from these programmable logic devices. An interface may include any suitable input/output component of a circuit or device. For example, the input/output component of memory, such as RAM, EPROM, ROM, or the like, the input/output components of programmable logic devices, other ~~than~~ integrated circuits, or any other suitable input/output component may be considered an interface for purposes of the present invention.

Please replace paragraph 22 with the following:

[0003] Transceiver 102 includes a transmit portion, corresponding to output 120, and a receive portion, corresponding to input 122. Test data stream 132 is transmitted via output 120 to input 122 of transceiver 102. The bit error rate of transceiver 102 is analyzed using the compare circuitry of bit error rate tester 104. The compare circuitry may include, for example, formatter 116, ~~comparer~~ comparator 114, and embedded memory 118.

Please replace paragraph 25 with the following:

[0004] Formatter 116 may be any suitable circuitry configured to format the incoming bit stream from transceiver 102. Formatter 116 may pad bits, remove bits, rearrange bits, or otherwise modify the incoming bit stream in any suitable way. For example, if the incoming bit stream is in a single word format and if ~~comparer~~ comparator 114 is configured to compare double words, formatter 116 may be configured to append consecutive words to form double word data to be used by ~~comparer~~ comparator 114.

Please replace paragraph 26 with the following:

[0005] In some embodiments of the present invention, formatter 116 may be unnecessary (e.g., when comparison data 134 is stored in a compatible format and when ~~comparer~~ comparator 114 is configured to accept the incoming bit stream in its present form). In some embodiments of the present invention, ~~comparer~~ comparator 114 may be configured to perform any requisite formatting operations to the incoming bit stream. In such embodiments, the functionality of formatter 116 may be accordingly modified or removed.

Please replace paragraph 27 with the following:

[0006] When the input bit stream is formatted (if necessary), the formatted input bit stream is transmitted to ~~comparer~~ comparator 114. ~~Comparer~~ Comparator 114 is any suitable circuitry configured to compare the incoming data to comparison data 134. ~~Comparer~~ Comparator 114 may maintain a counter that indicates how many errors occurred in the bits of the input data.

Please replace paragraph 28 with the following:

[0007] Bit error rate tester 104 works by sending test data 130 across transceiver 102 and to ~~comparer~~ comparator 114. This is done repetitively. That is, transmitter 102 accesses and communicates test data 130 continuously such that the bit stream being communicated over transceiver 102 is a data pattern made of the repeated values represented by test data 130 (and, if applicable, as formatted by formatter 106). When bit error rate testing begins, ~~comparer~~ comparator 114 may not be immediately aware at which value in the incoming data pattern ~~comparer~~ comparator 114 is looking. For example, if test data 130 and comparison data 134 are both the same pattern of data made of a certain number of values, when a first value comes into ~~comparer~~ comparator 114, ~~comparer~~ comparator 114 may not be aware to which value in comparison data 134 the incoming value should be compared. In order to synchronize ~~comparer~~ comparator 114, a process such as that shown in FIG. 2 may be used.

Please replace paragraph 29 with the following:

[0008] FIG. 2 is a flow-chart of illustrative steps involved in synchronizing bit error rate tester 104 for comparing incoming data from transceiver 102. The circuitry that implements ~~comparer~~ comparator 114 may be configured to process the steps illustrated in FIG. 2. For example, ~~comparer~~ comparator 114 may use an address identifier shown as the integral variable "addr" in FIG. 2. "Addr" represents the location of the value currently being compared in comparison data 134. "Addr" is initialized to the value "0" (i.e., corresponding to a first value) when ~~comparer~~ comparator 114 first begins the synchronization process at step 200.

Please replace paragraph 30 with the following:

[0009] At step 200, the value of comparison data 134 residing at address 0 is compared to the incoming data from transceiver 102. If there is a match, then at step 202, "addr" is increased by 1 and the value of comparison data 134 at address 1 is compared to the next piece of incoming data from transceiver 102. If there is another match, then the value of "addr" is increased by 1 again and the next piece of comparison data 134 is compared to the next incoming piece of data from transceiver 102. This may continue until ~~comparer~~ comparator 114 determines that the data being compared do not match.

Please replace paragraph 32 with the following:

[0010] To illustrate this process, if comparison data 134 is a data stream that contains the values a, b, c, d, e, f, g, h, i, j, k corresponding to "addr" values of 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, respectively, and if the incoming data from transceiver 102 is c, d, e, f, g, h, i, j, k, then the progression in values of "addr" would look like 0, 2, 4, 5, 6, 7, 8, 9, 10. This corresponds to the following values stored in comparison data 134 being compared to the incoming data (i.e., in this order): a, c, e, f, g, h, i, j, k. Thus, after the first two incoming values, ~~comparer~~ comparator 114 has synchronized itself to the incoming data coming from transceiver 102.

Please replace paragraph 34 with the following:

[0011] When bit error rate tester 104 has been synchronized, ~~comparer~~ comparator 114 begins comparing each incoming value to the respective value stored in comparison data 134. Whenever a non-match is detected, an error count is updated (e.g., increased by one). ~~Comparer~~ Comparator 114 may be configured to compare every bit received from transceiver 102 to comparison data 134 such that the bit error rate result is not merely based on a sampling of data. Alternatively, the ~~comparer~~ comparator 114 may be configured to compare only a sampling of incoming data to comparison data 134. The latter approach may be useful when, for example, a rough and quick estimate of the bit error rate is desired.

Please replace paragraph 37 with the following:

[0012] Controller 112 may be any suitable circuitry configured to process the instructions received from user equipment 124 and to control the execution of the individual components of bit error rate tester 104. For example, controller 112 may be responsible for generating and transmitting a start signal (i.e., to begin testing), a stop signal (i.e., to end testing), and a reset signal (i.e., to reset the components to an initial state). Controller 112 may also be responsible for collecting information regarding, for example, status of testing (e.g., number of errors), and whether the components have been successfully reset following a reset instruction. Controller 112 may be used to interface with user equipment 124. For example, controller 112 may interpret commands from user equipment, may communicate information regarding testing status, and may otherwise serve as an interface between the bit error ~~rates~~ rate testing operations of programmable logic device 100 and user equipment 124. These are merely illustrative operations for which controller 112 may be responsible. It will be understood that controller 112 may be used for any other suitable operations in addition to or in place of those described.